Package
com.jpbc

com.jpbc
Interface PBStatement

All Subinterfaces:
PdlStatement, PmlStatement, PqlStatement

public interface **PBStatement**
extends

The object used for executing a static PB statement

Method Summary

void	close() Releases this PBStatement object and JPBC resources immediately instead of waiting for this to happen when it is automatically closed.
java.lang.Object	execute() Executes the given PB statement.
java.lang.Object	execute(java.lang.String str) Executes the given statement.

Methods

execute

```
public java.lang.Object execute()
  throws PException
```

Executes the given PB statement.

Returns:
the execution output

execute

```
public java.lang.Object execute(java.lang.String str)
  throws PException
```

Executes the given statement.

Parameters:
str - the PSY statement to be executed

Returns:
the number of affected patterns

close

```
public void close()
  throws PException
```

(continued from last page)

Releases this `PBStatement` object and JPBC resources immediately instead of waiting for this to happen when it is automatically closed. It is generally good practice to release resources as soon as you are finished with them to avoid tying up database resources.

Calling the method `close` on a `PBStatement` object that is already closed has no effect.

com.jpbc
Interface PConnection

public interface **PConnection**
extends

A connection (session) with a specific PBMS.

Method Summary

void	close() Releases this PConnection object's pbms and JPBC resources immediately instead of waiting for them to be automatically released.
PBStatement	createPBStatement(java.lang.String command) Creates a PBStatement object for sending PB statements to the PBMS.
PdlStatement	createPDLStatement(java.lang.String command) Creates a PdlStatement object for sending PDL statements to the PBMS.
PmlStatement	createPMLStatement(java.lang.String command) Creates a PmlStatement object for sending PML statements to the PBMS.
PqlStatement	createPQLStatement(java.lang.String command) Creates a PqlStatement object for sending PQL statements to the PBMS.

Methods

createPBStatement

```
public PBStatement createPBStatement(java.lang.String command)
  throws PException
```

Creates a PBStatement object for sending PB statements to the PBMS.

Returns:
a new default PBStatement object

Throws:
PException - if a pbms access error occurs

createPDLStatement

```
public PdlStatement createPDLStatement(java.lang.String command)
  throws PException
```

Creates a PdlStatement object for sending PDL statements to the PBMS.

Returns:
a new default PdlStatement object

Throws:

(continued from last page)

PException - if a pbms access error occurs

createPMLStatement

```
public PmlStatement createPMLStatement(java.lang.String command)
  throws PException
```

Creates a `PmlStatement` object for sending PML statements to the PBMS.

Returns:

a new default `PmlStatement` object

Throws:

PException - if a pbms access error occurs

createPQLStatement

```
public PqlStatement createPQLStatement(java.lang.String command)
  throws PException
```

Creates a `PqlStatement` object for sending PQL statements to the PBMS.

Returns:

a new default `PqlStatement` object

Throws:

PException - if a pbms access error occurs

close

```
public void close()
  throws PException
```

Releases this `PConnection` object's pbms and JPBC resources immediately instead of waiting for them to be automatically released.

Calling the method `close` on a `PConnection` object that is already closed is a no-op.

Note: A `PConnection` object is automatically closed when it is garbage collected. Certain fatal errors also close a `PConnection` object.

com.jpbc
Interface PdlStatement

All Superinterfaces:
PBStatement

public interface **PdlStatement**
extends PBStatement

It specializes the `PBStatement` interface in order to support the creation (and deletion) of new entities within the PBMS

Method Summary

`java.lang.String`	`executePDLcommand()` Executes the given PDL statement.
`java.lang.String`	`executePDLcommand(java.lang.String str)` Executes the given PDL statement.

Methods inherited from interface `com.jpbc.PBStatement`

`close`, `execute`, `execute`

Methods

executePDLcommand

```
public java.lang.String executePDLcommand()
  throws PException
```

Executes the given PDL statement.

Returns:
A message containing the outcome of the execution

executePDLcommand

```
public java.lang.String executePDLcommand(java.lang.String str)
  throws PException
```

Executes the given PDL statement.

Parameters:
`str` - the PSY-PDL statement to be executed

Returns:
the number of affected patterns

com.jpbc
Interface PDriver

public interface **PDriver**
extends

The interface that every jpbc driver class must implement.

Method Summary

PConnection	connect(java.lang.String url, java.lang.String user, java.lang.String pwd) Attempts to make a PBMS connection to the given URL.
int	PBMSCompliant() Reports the compliance level of the underlying PBMS with respect to the general model and languages supported by JPBC.

Methods

connect

```
public PConnection connect(java.lang.String url,
                           java.lang.String user,
                           java.lang.String pwd)
  throws PException
```

Attempts to make a PBMS connection to the given URL. The driver should return "null" if it realizes it is the wrong kind of driver to connect to the given URL. This will be common, as when the JPBC driver manager is asked to connect to a given URL it passes the URL to each loaded driver in turn.

The driver should throw an `PException` if it is the right driver to connect to the given URL but has trouble connecting to the PBMS.

Parameters:

`url` - the URL of the PBMS to which to connect
`user` - the user name
`pwd` - the user password

Returns:

a `PConnection` object that represents a connection to the PBMS

Throws:

`PException` - if a PBMS access error occurs

PBMSCompliant

```
public int PBMSCompliant()
```

Reports the compliance level of the underlying PBMS with respect to the general model and languages supported by JPBC. We remark that 0-compliant drivers can be implemented for most pattern management solutions provided by DBMSs vendors.

Returns:

the compliance level of the underlying PBMS

com.jpbc
Class PDriverManager

```
java.lang.Object
   |
   +-com.jpbc.PDriverManager
```

public class **PDriverManager**
extends java.lang.Object

Similarly to what happen in JDBC, the `PDriverManager` class manages the list of available drivers and establishes connections to a PBMS using a certain driver. A driver is available for accessing a certain PBMS only after it has been loaded.

Method Summary

static PConnection	getConnection(java.lang.String url, java.lang.String user, java.lang.String pwd) Creates a connection with a PBMS, using a pre-registered driver.
static void	registerDriver(PDriver driver) Registers the given driver with the `PDriverManager`.

Methods inherited from class `java.lang.Object`

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods

getConnection

```
public static PConnection getConnection(java.lang.String url,
         java.lang.String user,
         java.lang.String pwd)
  throws PException
```

Creates a connection with a PBMS, using a pre-registered driver.

Parameters:

`url` - a location string identifying the PBMS
`user` - the account user name
`pwd` - the account password

registerDriver

```
public static void registerDriver(PDriver driver)
  throws PException
```

Registers the given driver with the `PDriverManager`. A newly-loaded driver class should call the method `registerDriver` to make itself known to the `PDriverManager`.

Parameters:

`driver` - the new JPBC Driver that is to be registered with the `PDriverManager`

(continued from last page)

Throws:

`PException` - if an access error occurs

com.jpbc
Class PException

```
java.lang.Object
   |-java.lang.Throwable
      |-java.lang.Exception
         |-com.jpbc.PException
```

All Implemented Interfaces:
java.io.Serializable

---

public class **PException**
extends java.lang.Exception

An exception that provides information on a PBMS access error or other errors.

Constructor Summary

public	PException(java.lang.String reason, java.lang.String errorCode) Constructs a fully-specified PException object.
public	PException(java.lang.String reason) Constructs an PException object with a reason;

Method Summary

java.lang.String	getErrorCode() Retrieves the exception code for this PException object.

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

Constructors

PException

```
public PException(java.lang.String reason,
                  java.lang.String errorCode)
```

Constructs a fully-specified PException object.

Parameters:
reason - a description of the exception
errorCode - the exception's code

PException

`public PException(java.lang.String reason)`

Constructs an `PException` object with a reason;

Parameters:

`reason` - a description of the exception

Methods

getErrorCode

`public java.lang.String getErrorCode()`

Retrieves the exception code for this `PException` object.

Returns:

the exception error code

com.jpbc
Interface PmlStatement

All Superinterfaces:
PBStatement

---

public interface **PmlStatement**
extends PBStatement

It specializes the `PBStatement` interface in order to support the manipulation of PBMS elements.

Method Summary

`int`	`executePMLcommand()` Executes the given PML statement.
`int`	`executePMLcommand(java.lang.String str)` Executes the given PML statement.

Methods inherited from interface `com.jpbc.PBStatement`

`close`, `execute`, `execute`

Methods

executePMLcommand

```
public int executePMLcommand()
  throws PException
```

Executes the given PML statement.

Returns:
the number of affected patterns

---

executePMLcommand

```
public int executePMLcommand(java.lang.String str)
  throws PException
```

Executes the given PML statement.

Parameters:
`str` - the PSY-PML statement to be executed

Returns:
the number of affected patterns

com.jpbc
Interface PqlStatement

All Superinterfaces:
PBStatement

public interface **PqlStatement**
extends PBStatement

It specializes the `PBStatement` interface in order to allow any Java application to execute queries against the PBMS for which a connection has been created

Method Summary

`PResultSet`	`executePQLcommand()` Executes the given PQL statement.
`PResultSet`	`executePQLcommand(java.lang.String str)` Executes the given PQL statement.

Methods inherited from interface `com.jpbc.PBStatement`

`close`, `execute`, `execute`

Methods

executePQLcommand

```
public PResultSet executePQLcommand()
  throws PException
```

Executes the given PQL statement.

executePQLcommand

```
public PResultSet executePQLcommand(java.lang.String str)
  throws PException
```

Executes the given PQL statement.

Parameters:
`str` - the PSY-PQL statement to be executed

Returns:
the number of affected patterns

com.jpbc
Interface PResultSet

public interface **PResultSet**
extends

The `PResultSet` interface provides methods for managing the result of a PQL query execution. In particular, the `PResultSet` interface defines a next method used to scroll forward a `PResultSet` object, i.e., to move the cursor forward over the query result, and a method `getXXX` methods, allowing the application to access the various attributes of the result set, identified both by name or by positions, and to convert their content in an instance of the Java type XXX (thus, JPBC supports type coercion). In particular, when the query returns a set of patterns, which are indeed objects, the method getObject has to be used, in order to translate the value contained in the indicated column into a Java Object. The object components can then be accessed by using the getAttributes method.

Method Summary

`void`	`close()` Releases this `PResultSet` object and JPBC resources immediately instead of waiting for this to happen when it is automatically closed.
`java.io.InputStream`	`getAsciiStream(int columnIndex)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a stream of ASCII characters.
`java.io.InputStream`	`getAsciiStream(java.lang.String columnName)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a stream of ASCII characters.
`java.io.InputStream`	`getBinaryStream(int columnIndex)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a binary stream of uninterpreted bytes.
`java.io.InputStream`	`getBinaryStream(java.lang.String columnName)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a stream of uninterpreted `bytes`.
`boolean`	`getBoolean(int columnIndex)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a `boolean` in the Java programming language.
`boolean`	`getBoolean(java.lang.String columnName)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a `boolean` in the Java programming language.
`byte`	`getByte(int columnIndex)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` in the Java programming language.
`byte`	`getByte(java.lang.String columnName)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` in the Java programming language.
`byte[]`	`getBytes(int columnIndex)` Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` array in the Java programming language.

byte[]	getBytes(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a byte array in the Java programming language.
java.sql.Date	getDate(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a java.sql.Date object in the Java programming language.
java.sql.Date	getDate(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a java.sql.Date object in the Java programming language.
double	getDouble(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a double in the Java programming language.
double	getDouble(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a double in the Java programming language.
float	getFloat(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a float in the Java programming language.
float	getFloat(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a float in the Java programming language.
int	getInt(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as an int in the Java programming language.
int	getInt(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as an int in the Java programming language.
long	getLong(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a long in the Java programming language.
long	getLong(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a long in the Java programming language.
java.lang.Object	getObject(int columnIndex) Gets the value of the designated column in the current row of this PResultSet object as an Object in the Java programming language.
short	getShort(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a short in the Java programming language.
short	getShort(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a short in the Java programming language.
java.lang.String	getString(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a String in the Java programming language.

java.lang.String	getString(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a String in the Java programming language.
java.sql.Time	getTime(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a java.sql.Time object in the Java programming language.
java.sql.Time	getTime(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a java.sql.Time object in the Java programming language.
java.sql.Timestamp	getTimestamp(int columnIndex) Retrieves the value of the designated column in the current row of this PResultSet object as a java.sql.Timestamp object in the Java programming language.
java.sql.Timestamp	getTimestamp(java.lang.String columnName) Retrieves the value of the designated column in the current row of this PResultSet object as a java.sql.Timestamp object.
boolean	next() Moves the cursor down one row from its current position.

Methods

next

```
public boolean next()
  throws PException
```

Moves the cursor down one row from its current position. A PResultSet cursor is initially positioned before the first row; the first call to the method next makes the first row the current row; the second call makes the second row the current row, and so on.

If an input stream is open for the current row, a call to the method next will implicitly close it. A PResultSet object's warning chain is cleared when a new row is read.

Returns:

true if the new current row is valid; false if there are no more rows

Throws:

PException - if a dbms access error occurs

close

```
public void close()
  throws PException
```

Releases this PResultSet object and JPBC resources immediately instead of waiting for this to happen when it is automatically closed.

getString

```
public java.lang.String getString(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this PResultSet object as a String in the Java programming language.

(continued from last page)

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:
`PException` - if a database access error occurs

---

getBoolean

```
public boolean getBoolean(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `boolean` in the Java programming language.

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `false`

Throws:
`PException` - if a database access error occurs

---

getByte

```
public byte getByte(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` in the Java programming language.

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

---

getShort

```
public short getShort(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `short` in the Java programming language.

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

getInt

```
public int getInt(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as an `int` in the Java programming language.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:

`PException` - if a database access error occurs

getLong

```
public long getLong(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `long` in the Java programming language.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:

`PException` - if a database access error occurs

getFloat

```
public float getFloat(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `float` in the Java programming language.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:

`PException` - if a database access error occurs

getDouble

```
public double getDouble(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `double` in the Java programming language.

(continued from last page)

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

---

getBytes

```
public byte[] getBytes(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` array in the Java programming language. The bytes represent the raw values returned by the driver.

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:
`PException` - if a database access error occurs

---

getDate

```
public java.sql.Date getDate(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `java.sql.Date` object in the Java programming language.

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:
`PException` - if a database access error occurs

---

getTime

```
public java.sql.Time getTime(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `java.sql.Time` object in the Java programming language.

Parameters:
`columnIndex` - the first column is 1, the second is 2, ...

Returns:
the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:
`PException` - if a database access error occurs

getTimestamp

```
public java.sql.Timestamp getTimestamp(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `java.sql.Timestamp` object in the Java programming language.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:

`PException` - if a database access error occurs

getAsciiStream

```
public java.io.InputStream getAsciiStream(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a stream of ASCII characters. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large LONGVARCHAR values. The JDBC driver will do any necessary conversion from the database format into ASCII.

Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return `0` when the method `InputStream.available` is called whether there is data available or not.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

a Java input stream that delivers the database column value as a stream of one-byte ASCII characters; if the value is SQL `NULL`, the value returned is `null`

Throws:

`PException` - if a database access error occurs

getBinaryStream

```
public java.io.InputStream getBinaryStream(int columnIndex)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a binary stream of uninterpreted bytes. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large `LONGVARBINARY` values.

Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return `0` when the method `InputStream.available` is called whether there is data available or not.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

a Java input stream that delivers the database column value as a stream of uninterpreted bytes; if the value is SQL `NULL`, the value returned is `null`

Throws:

(continued from last page)

`PException` - if a database access error occurs

getString

```
public java.lang.String getString(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `String` in the Java programming language.

Parameters:

`columnName` - the name of the column

Returns:

the column value; if the value is `NULL`, the value returned is `null`

Throws:

`PException` - if a pattern base access error occurs

getBoolean

```
public boolean getBoolean(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `boolean` in the Java programming language.

Parameters:

`columnName` - the name of the column

Returns:

the column value; if the value is SQL `NULL`, the value returned is `false`

Throws:

`PException` - if a database access error occurs

getByte

```
public byte getByte(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` in the Java programming language.

Parameters:

`columnName` - the SQL name of the column

Returns:

the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:

`PException` - if a database access error occurs

getShort

```
public short getShort(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `short` in the Java programming language.

(continued from last page)

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

---

getInt

```
public int getInt(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as an `int` in the Java programming language.

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

---

getLong

```
public long getLong(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `long` in the Java programming language.

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

---

getFloat

```
public float getFloat(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `float` in the Java programming language.

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

getDouble

```
public double getDouble(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `double` in the Java programming language.

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `0`

Throws:
`PException` - if a database access error occurs

getBytes

```
public byte[] getBytes(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `byte` array in the Java programming language. The bytes represent the raw values returned by the driver.

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:
`PException` - if a database access error occurs

getDate

```
public java.sql.Date getDate(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `java.sql.Date` object in the Java programming language.

Parameters:
`columnName` - the SQL name of the column

Returns:
the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:
`PException` - if a database access error occurs

getTime

```
public java.sql.Time getTime(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `java.sql.Time` object in the Java programming language.

(continued from last page)

Parameters:

`columnName` - the SQL name of the column

Returns:

the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:

`PException` - if a database access error occurs

---

getTimestamp

```
public java.sql.Timestamp getTimestamp(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a `java.sql.Timestamp` object.

Parameters:

`columnName` - the SQL name of the column

Returns:

the column value; if the value is SQL `NULL`, the value returned is `null`

Throws:

`PException` - if a database access error occurs

---

getAsciiStream

```
public java.io.InputStream getAsciiStream(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a stream of ASCII characters. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large `LONGVARCHAR` values. The JDBC driver will do any necessary conversion from the database format into ASCII.

Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return `0` when the method `available` is called whether there is data available or not.

Parameters:

`columnName` - the SQL name of the column

Returns:

a Java input stream that delivers the database column value as a stream of one-byte ASCII characters. If the value is SQL `NULL`, the value returned is `null`.

Throws:

`PException` - if a database access error occurs

---

getBinaryStream

```
public java.io.InputStream getBinaryStream(java.lang.String columnName)
  throws PException
```

Retrieves the value of the designated column in the current row of this `PResultSet` object as a stream of uninterpreted `bytes`. The value can then be read in chunks from the stream. This method is particularly suitable for retrieving large `LONGVARBINARY` values.

Note: All the data in the returned stream must be read prior to getting the value of any other column. The next call to a getter method implicitly closes the stream. Also, a stream may return `0` when the method `available` is called whether there is data available or not.

(continued from last page)

Parameters:

`columnName` - the SQL name of the column

Returns:

a Java input stream that delivers the database column value as a stream of uninterpreted bytes; if the value is SQL `NULL`, the result is `null`

Throws:

`PException` - if a database access error occurs

---

getObject

```
public java.lang.Object getObject(int columnIndex)
  throws PException
```

Gets the value of the designated column in the current row of this `PResultSet` object as an `Object` in the Java programming language.

This method may also be used to read database-specific abstract data types.

Parameters:

`columnIndex` - the first column is 1, the second is 2, ...

Returns:

a `java.lang.Object` holding the column value

Throws:

`PException` - if an access error occurs

com.jpbc
Interface PStruct

public interface **PStruct**
extends

The standard mapping in the Java programming language for a structured type. A `PStruct` object contains a value for each attribute of the structured type that it represents. By default, an instance of`PStruct` is valid as long as the application has a reference to it.

Method Summary

`java.lang.Object[]`	`getAttributes()` Produces the ordered values of the attributes of the structured type that this `PStruct` object represents.
`java.lang.String`	`getTypeName()` Retrieves the type name that this `Struct` object represents.

Methods

getTypeName

```
public java.lang.String getTypeName()
```

Retrieves the type name that this `Struct` object represents.

Returns:
the fully-qualified type name of the structured type for which this `PStruct` object is the generic representation

Throws:
`PException` - if a pbms access error occurs

getAttributes

```
public java.lang.Object[] getAttributes()
  throws PException
```

Produces the ordered values of the attributes of the structured type that this `PStruct` object represents. This method uses the type map associated with the connection for customizations of the type mappings. If there is no entry in the connection's type map that matches the structured type that this `PStruct` object represents, the driver uses the standard mapping.

Conceptually, this method calls the method `getObject` on each attribute of the structured type and returns a Java array containing the result.

Returns:
an array containing the ordered attribute values

Throws:
`PException` - if an error occurs